KING SAUD UNIVERSITY
COLLEGE OF COMPUTER & INFORMATION SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Course: Algorithms Design and Analysis - CSC311
Semester: Spring 2023
Instructors: Prof. Mohamed Maher Ben Ismail & Prof. Aqil Azmi

Theoretical and Empirical Analysis of Huffman Encoding and Decoding Algorithms

DEADLINE: ***June 1^{sh} 2023, 23:59****.*

1. Introduction

Huffman Encoding is one of the simplest algorithms to compress data. Even though it is very old and simple, it is still widely used (eg: in few stages of JPEG, MPEG, etc). In this project you will implement Huffman encoding and decoding. Your system must accept a file and (Huffman) tree for the same. Use the priority queue to select nodes with smallest frequencies and construct the tree. Once you have constructed the tree, traverse the tree and create a dictionary of codewords (letter to code). Given any new sentences, your system must show how the sentence is converted to Huffman code and then decoded back to original sentence. Note that you must implement BST and Heap yourself and must not rely on any language libraries. You can use external libraries like Graph Viz to display your Huffman tree.

You will be working in a team of **two** students to fulfill the project requirements.

2. Project Goals

The intended goals of this project include:

- Acquisition of knowledge about Huffman encoding and decoding algorithms.
- Programming of the considered algorithms.
- Conducting computational experiments.
- Interpretation of the obtained results.
- Writing a technical report.

3. Experiments

Experiments have to be carried out according to the following directions.

1. Each algorithm has to be run on test data. Data will be generated in a random way.
2. If necessary, an algorithm may be run on test data a fixed number of times so that the running times obtained are meaningful
3. Running times will be plotted against input size.

4. Programming

Implementation of algorithms may be done in any language of student's choice. However, the language and its compiler should support certain features in order to be able to run the experiments properly. The choice of C, C++, Java, Python, Matlab or the like should be enough. Source code has to be handed over.

5. Project Demonstration

Once the project is completed, the following is expected from you:

a) A demonstration of your project in which you show the various features of your system such as its correctness, efficiency, etc. You should be prepared to answer detailed questions on the system design and implementation during this demo. We will also examine your code to check for code quality, code documentation, etc.
b) You should hand in a completed project report which contains details about your project, such as main data structures, main components of the algorithm, design of the user-interface for input/output, experimental results, e.g. charts of running time versus input size, etc.
c) You should also turn in your code and associated documentation (e.g. README files) so that everything can be backed up for future reference.
d) Email your code and all associated files with "CSC311-Project <Lastname>" as subject.

6. Written Report

A report describing the following points must be handed over.

• Brief explanation of the algorithms.
• Brief explanation of the implementations. It can be done by including sufficiently detailed comments in the code.
• Brief description of the experiment.

• Interpretation of experimental data. Comparison of experimental data with theoretical complexities.
• Conclusions. In this section students must draw their own conclusions (be creative).

The report has to be written in correct English; it also has to possess clarity of thought. Show me what you know; do not force to search for it through a poorly written paper.

7. Grading

We will take points off when:
• There is spelling mistakes
• It is plenty of irrelevant material. Down with the irrelevant!
• It lacks clarity of thought.
• It is lengthy, long-winded or poor in content.
• Code is not properly commented.
• Code is not properly structured.
• Variables have absurd names.
• There are run-time errors.

8. Questions and Office Hours

Mr. Wesam Hatamleh, Prof. Aqil Azmi and Prof. Mohamed Maher ben Ismail are willing to answer your questions about algorithms, complexity or the experiment. They will not answer questions about coding errors as it is our feeling that, at this point, writing error-free code is your responsibility.